N90-23063

Man-in-the-control-loop Simulation of Manipulators

J. L. Chang, T. C. Lin, and K. H. Yae

Center for Simulation and Design Optimization
Department of Mechanical Engineering
The University of Iowa
Iowa City, Iowa 52240

ABSTRACT

A method to achieve man-in-the-control-loop simulation is presented. Emerging real-time dynamics simulation suggests a potential for creating an interactive design workstation with a human operator in the control loop. The recursive formulation for multibody dynamics simulation is studied to determine requirements for man-in-the-control-loop simulation. High speed computer graphics techniques provides realistic visual cues for the simulator. Backhoe and robot arm simulations are implemented to demonstrate the capability of man-in-the-control-loop simulation.

1. INTRODUCTION

Man-in-the-control-loop simulation has been used in vehicle design and pilot training. The first visual vehicle simulation was done by Sheridan, Paynter, and Coons in 1964 [1]. In the early 1970's, Volkswagen built a motion-base driving simulator to test the influence of vehicle parameters on safety [2]. General Motors had its own fixed-base driving simulator to test driver-vehicle performance in the late 1970's [3-5]. Daimler-Benz built an advanced driving simulator in 1984 [6-7]. This system has a 180-degree-wide image on a dome, and its motion platform allows a driver to use his peripheral vision in a natural way and to make judgements about handling qualities of new vehicle designs.

The flight simulator arrived at its modern form at the end of the 1960's. Fixed-base simulators with simple dynamics models were established before then. The flight simulators have assumed an increasingly crucial role in the development and testing of new product lines in the aviation industry. For instance, the Boeing Flight Systems Laboratory has its advanced flight simulators [8]. They are used to design, develop, and validate a newly developed aircraft as well as to train pilots.

Man-in-the-control-loop simulation has an enormous potential for dynamic system design. The simulator provides realistic environmental cues to allow the operator to control the simulated system, as he would in the real world. This simulator provides a very natural and realistic simulation environment for the teleoperation. It also provides a low-cost environment for man/machine interaction, thus leading us to human factor research. With the recent development of a highly efficient multibody dynamic modelling method, a quantum leap in computer hardware and software, and a substantial cost reduction in high-speed computer graphics, the real-time man-in-the-control-loop simulation appears to be within our reach.

In comparison to the conventional design procedure that typically requires a long development time from a prototype to the final product, "simulation-aided design" appears to be an efficient design tool that would make it possible that the dynamic characteristics of a system are predicted in the early design stage. The designer can observe the behavior of a mechanical system from the simulation and can correct any design flaw even before the fabrication and test of prototypes. At the same time, the operator can provide valuable feedback to the designers about the system's characteristics that require further improvement. Therefore, the quality of the system can be improved and the cost of prototype fabrication and test can be reduced. In this way, the machine can be ergonomically and economically designed to be adapted to the human, instead of forcing the human to adapt to the machine after the final design.

A low-cost network-based simulator is used in this research. In this simulator, the dynamic simulator in a parallel-processing computer is integrated with high-speed computer graphics through the Network Computing System (by Apollo Computer, Inc.). And the operator comes in the loop through a serial port in the IBM PC that digitizes operator's control action on the joysticks. Here, we are focusing on two special systems: backhoe and robot arm. The backhoe has four degrees of freedom with a human operator in the loop all the time. Thus it makes a perfect example for this research. The same technique is applied to teleoperation in which the seven degree-of-freedom robot arm is controlled by the human operator in the loop.

2. METHODOLOGY

The real-time dynamic simulation demands a more efficient and accurate mathematical formulation. The recent development in the recursive formulation, in which the relative coordinates are used to yield a minimal set of differential and algebraic equation, makes the computation much more efficient when implemented with the parallel-processing algorithms. The numerical results from the simulation then have to be displayed in real-time. The quality of the graphics at the same time must be high enough to provide the human operator with high-fidelity realism. The operator's workstation that encompasses a host computer, a graphics computer, a control console, and interfaces between each other is introduced in this section.

2.1 NETWORK-BASED SIMULATOR

Figure 1 shows the setup of the network-based simulation. Two computers, an Alliant eight-processor computer and an Iris silicon graphics computer, are used in this simulation. They communicate through the Network Computing System(NCS). We use real-time dynamics formulation and Visualization of Dynamics System(VDS) to simulate and display the system. The dynamics code is in the Alliant/FX8 and the graphics package is in the Iris graphics workstation. The operator gives the control input by using a pair of joysticks, thus initiating the simulation. The joystick interface samples and digitizes the signal from the joysticks and sends it to VDS through serial communication. VDS sends the input command to the dynamics program. Then the dynamics program simulates the system and sends the updated position and orientation to VDS. VDS displays graphics either on the 19" Iris screen or on a large projection screen. Finally, the operator gets the visual feedback and issues a new command to complete the simulation cycle. In this application, a pair of joysticks is used to drive the mechanical systems. For the backhoe, each joystick has two axis controllers. The two joysticks have four controllers to drive four degrees of freedom of the backhoe. For the robot arm, each joystick has three axis controllers. Thus, six degrees of freedom of the robot arm can be driven by two joysticks.

2.2 DYNAMICS MODELLING

The absolute coordinate and relative coordinate are used to derive equations of motion for mechanical systems; however, the absolute coordinate is not suitable for high speed simulation due to its inefficiency. The recursive formulation of relative coordinates is applied to derive equations of motion to achieve high speed simulation. Haug and McCullough developed a systematical approach to derive equations of motion by using a variational-vector calculus formulation [9]. Bae and Haug employed the cut joint method and variational equations of motion to derive the recursive Newton-Euler equations of motion for constrained mechanical systems [10-11]. Recently, Bae, Hwang, and Haug refined this work by introducing a state vector notation [12]. This approach simplified the derivation and reduced the computing time for numerical simulation in certain classes of applications. Thus it is suitable for real-time simulation. The resulting form of the equations of motion from recursive formulation can be expressed as

$$\begin{bmatrix} \mathbf{M} & \mathbf{\Phi_q}^T \\ \mathbf{\Phi_q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{Q} \\ \boldsymbol{\gamma} \end{bmatrix} \qquad (1)$$

where the mass matrix $\mathbf{M}$ is obtained from the inertia properties of bodies and $\mathbf{\Phi_q}$ is Jacobian submatrices of cut joint constraints. $\mathbf{Q}$ is the vector that evaluates the inertia force and external force, and $\boldsymbol{\gamma}$ is the right side of constraint acceleration equations. A linear solver is used to obtain $\ddot{\mathbf{q}}$ that includes the acceleration vector of base body and the relative joint accelerations, and the Lagrange multipliers $\boldsymbol{\lambda}$. Cartesian accelerations of bodies can be recovered forward from the base body to tree end bodies. Relative joint velocities and accelerations are numerically integrated to obtain relative joint coordinates and velocities for the next time step.

The joint relative coordinate is used to derive the equations of motion for the backhoe and robot arm. The backhoe is the first application. For this given model, we chose swing tower, boom, dipper, and bucket as major components. The joint definitions and local vectors are shown in Figure 2. This backhoe is driven by hydraulic actuators. The computer model of robot arm is shown in Figure 3. The robot arm has seven major components and seven joints. Four of the seven joints are roll joints, the other three are pitch joints. So we have roll-pitch combinations. We used the same technique as in the backhoe; the only difference is that the robot arm is driven by a servo motor and harmonic drive.

The execution time is the major concern because of the real-time constraint. Figure 4 shows the relationship between execution time per one function evaluation and the number of processors for backhoe simulation. The straight line shows the largest integration stepsize. Below this line, the execution time is faster than real-time. The execution time for two processors is very close to real-time. Therefore, if we use more than three processors, we can achieve real-time or even faster than real-time. Meanwhile, fine grain parallelism is used to tune the dynamics program. The utility factor for four processors is 60%, but it drops to 40% for eight processors. That is because the backhoe system is too small: even though we assigned eight processors to the simulation, they were not fully utilized.

2.3 COMPUTER IMAGE GENERATION

The computer image generation plays an important role in man-in-the-control-loop simulation. Since computer graphics generates environmental cues, it should be realistic to the extent that the operator could have a similar experience with the simulator as he would with the actual system. It is required to provide the operator with the essential visual information in the simulator. The visual information should include a general perspective view of system, color of objects, surface texture, shading, and lighting. The graphics animation displays spatial position and orientation of bodies in a system at a high frame rate. Recently, the low cost simulation goal has become more realizable, owing to hardware and software improvement in graphics computers. Dubetz, Kuhl, and Haug presented an approach for interactively animated graphics for real-time dynamic simulation [13]. They implemented a network between workstation and host computer to display simulation graphics both in an interactive way and in a batch mode. Dubetz developed an interactive graphics package called the Visualization of Dynamic Systems that is capable of animating three-dimensional multibody systems with real-time rendering, viewing, and lighting operations [14].

VDS is used to generate realistic three-dimensional graphics either in batch mode or in interactive mode for man-in-the-control-loop simulation purposes. Three groups of data support VDS to display graphics: they are geometric data, visualization data, and system state data. Geometry data is used to describe geometry of components in their own fixed reference frame. Visualization data defines light, color, shading, texture, center of projection, view reference point, and so on. Both are pre-defined once for one system. System state data is defined for each frame, so it should contain the updated position and orientation of components in the interactive display. In the batch mode, we have to first create all the frames and then display. In the interactive mode, we create and display one frame after another. Parameters that are modified at a VDS interface and sent to a simulation server are called "valuators". Parameters that are modified by the simulation server and set to a VDS interface are called "displays". The valuators are controlled by the interface that VDS provides, such as knobs, dials, and mouse. In this way, VDS allows the user to display or control the simulated system interactively.

Using the batch-mode VDS, we obtained the realistic computer graphics of the backhoe. It has fourteen hundred polygons and eighteen bodies. Then we found that the display speed was too slow, about seven frames per second. For high speed display purposes, we reduced polygons to five hundred. As a result, we were able to display ten frames per second on Iris 4D/70 and twenty frames per second on the upgraded Iris, which has two processors.

In this research, VDS is the main controller for data communication and graphics display. VDS controls the operator's input, dynamics simulation, and graphics display. A couple of pictures from the interactive graphics display are shown in Figure 5. The pictures show that the backhoe pick up and dump an object. That is the quality we had in the interactive simulation. However, slow graphics display and network time delay made it difficult at this point to assess the realism of the simulated operation.

3. SUMMARY

The recursive dynamic formulation with parallel processing algorithms is capable of simulating dynamics of a mechanical system in real-time or sometimes even faster than real-time. But when it is integrated with high-fidelity graphics and an operator in the loop, the overall performance is not up to real-time yet. The slowdown was mostly caused by slow graphics display and communication time delay. To resolve these problems, the structure of the simulation is changed from network-based to workstation-based as shown in Figure 6. This new setup with one parallel-processing computer that houses both the dynamic simulation and graphics will be eventually able to eliminate the network time delay. And the visualization system is continually upgraded to increase display speed. The simulation cycle of the new setup is similar to that of the workstation-based simulation, except that the dynamic simulation, instead of VDS, controls the overall process.

In addition, the realism of the simulator will be enhanced by adding different types of feedback such as motion, auditory, and tactile feedback. When the actual operator console is linked up with the backhoe simulation and the Kraft mini-master with the robot arm simulators these simulators will possess a great potential for human factor research.

REFERENCES

1. Sheridan, T. B., Paynter, H. M., and Coons. S. A., "Some Novel Display Techniques for Driving Simulation," IEEE Transaction on Human Factor in Electronics, Vol. HFE-5, pp. 29-32, September 1964.

2. Richter, B., "Driving Simulator Studies: The Influence of Vehicle Parameters in Safety in Critical Situations," SAE Technical Paper, No. 741105.

3. Wierwille, W. W., "Driving Simulator Design for Realistic Handling," Proceedings of the Third International Conference on Vehicle System Dynamics, Swets and Zeitlinger, B. V., Amsterdam, 1975, pp. 186-199.

4. Repa, B. S. and Wierwille, W. W., "Driver Performance in Controlling a Driving Simulator with Varying Vehicle Response Characteristics," SAE Technical Paper, NO. 760779, October 1976.

5. Repa, B. S., Alexandridis, A. A. and Howell, L. J., "Study of Vehicle Steering and Response Characteristics in Simulated and Actual Driving," SAE Technical Paper, No. 780011, 1978.

6. Drosdol, J., and Panik, F., "The Daimler-Benz Driving Simulator: A tool for Vehicle Development," SAE Technical Paper, No. 850334, 1985.

7. Hahn, S., and Kalb. E., "The Daimler-Benz Driving Simulator Set-Up and Results of First Experiments," 1987 Summer Computer Simulation Conference Proceeding, pp. 993-998, 1987.

8. Fraser, T. L., and Phillips, C. E., "Engineering Flight Simulation-A Revolution of Change", SAE Technical Paper, No. 851901.

9. Haug, E. J., and McCullough, M. K., "A Variational-Vector Calculus Approach to Machine Dynamics," Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 108, pp. 25-30, March 1986.

10. Bae, D. S., and Haug, E. J., "A Recursive Formulation for Constrained Mechanical Systems, Part I - Open Loop," Mechanics of Structures and Machines, Vol. 15(3), pp. 359-382, 1987.

11. Bae, D. S., and Haug, E. J., "A Recursive Formulation for Constrained Mechanical Systems, Part II - Closed Loop," Mechanics of Structures and Machines, Vol. 15(4), 1987.

12. Bae, D. S., Hwang, R. S., and Haug, E. J., "A Recursive Formulation for Real-Time Dynamic Simulation," submitted to Journal of Mechanisms, Transmissions, and Automation in Design.

13. Dubetz, M. W., Kuhl, J. G., and Haug, E. J., "A Network Implementation of Real-Time Dynamic Simulation with Interactive Animated Graphics," submitted to Journal of Mechanisms, Transmissions, and Automation in Design.

14. Dubetz, M. W. , et al., User Documentation for Visualization of Dynamic System, Version 1.0, Center for Simulation and Design Optimization, The University of Iowa, 1988.

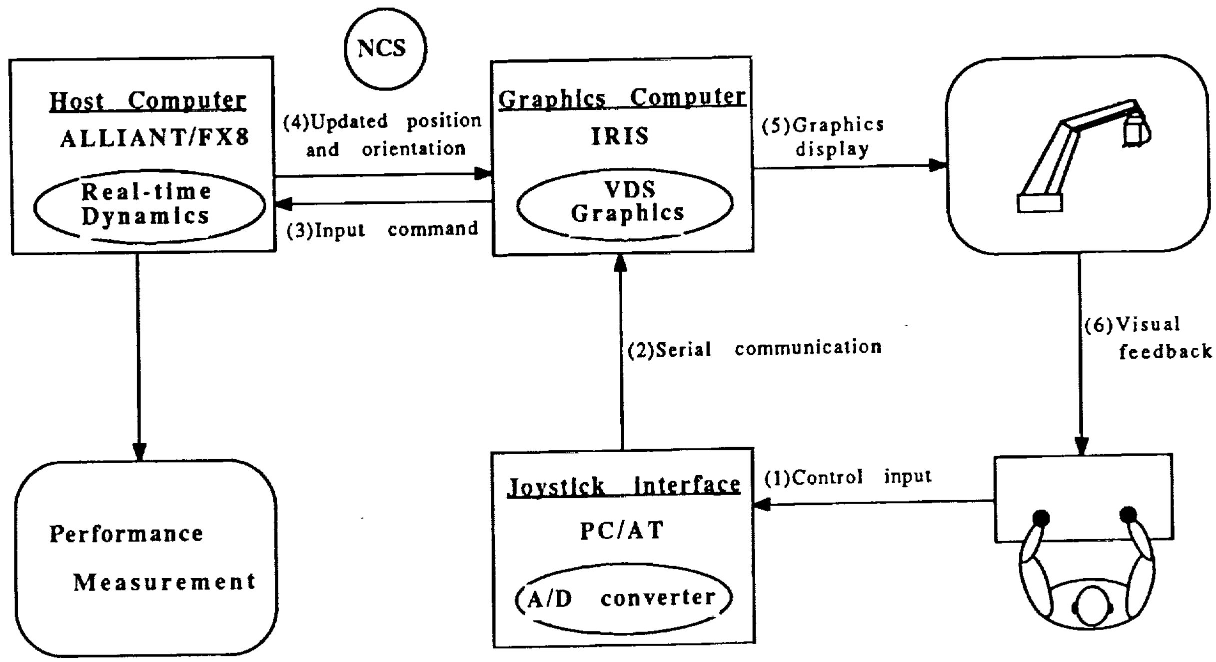

Figure 1. Network-based simulator

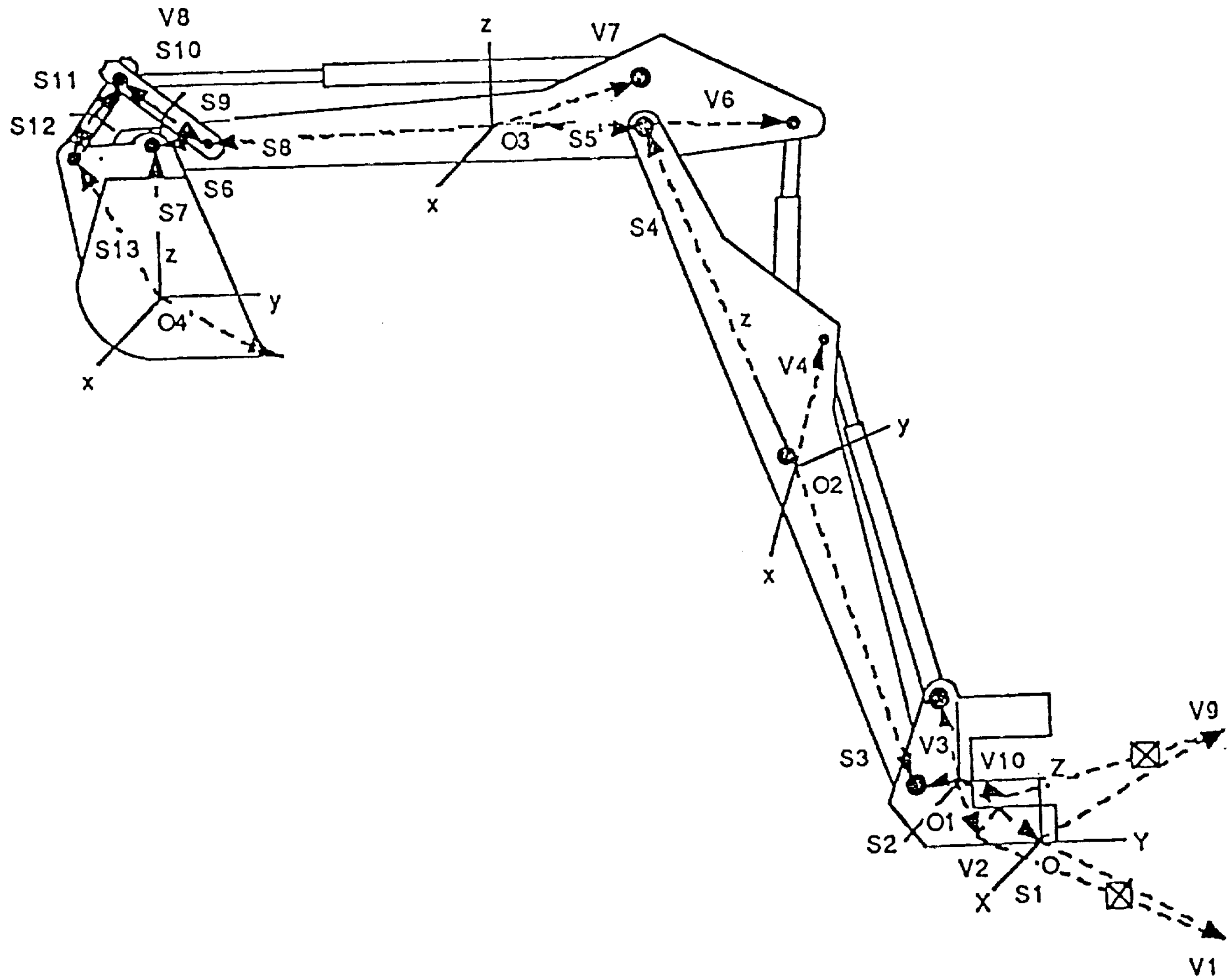

Figure 2. Computer model of backhoe

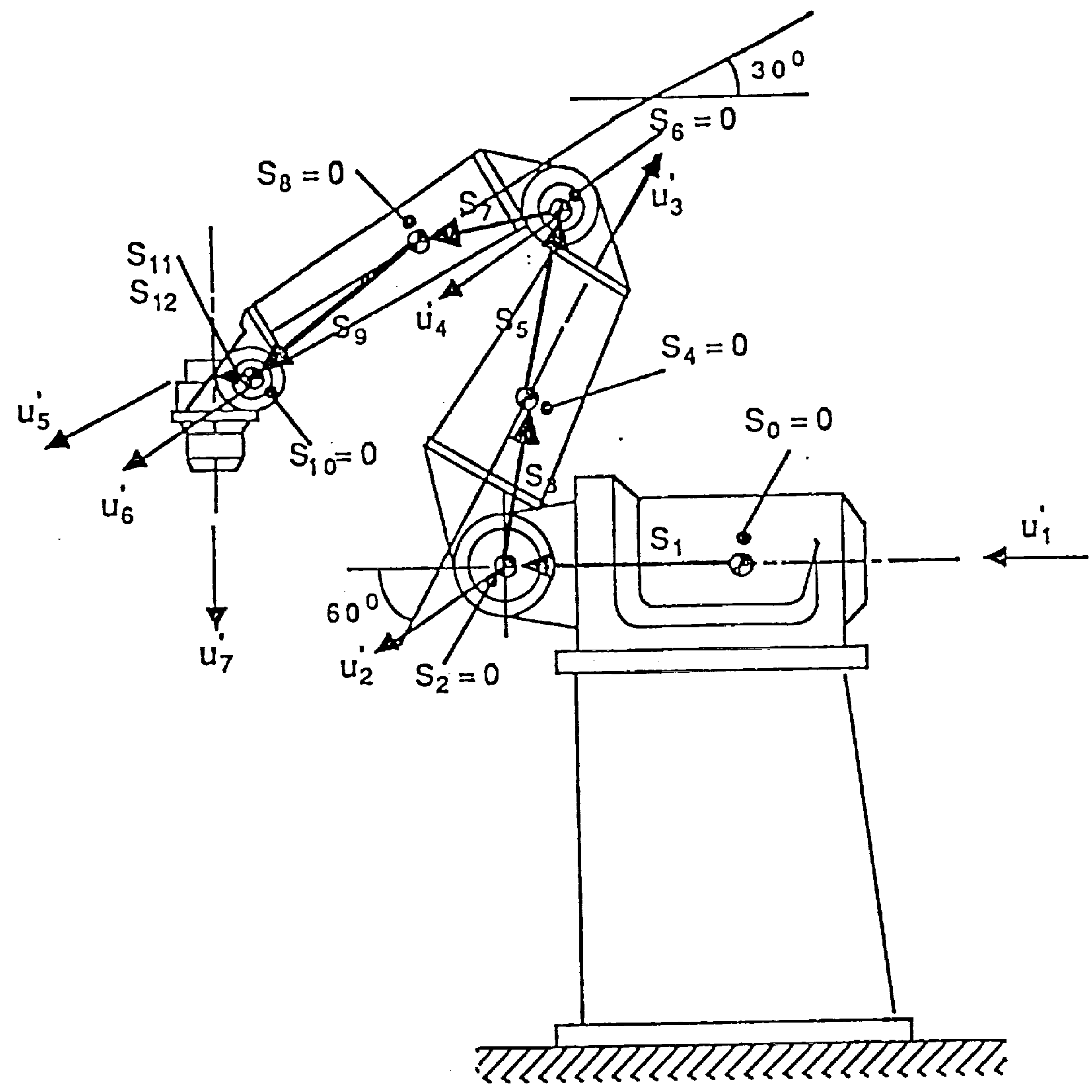

Figure 3. Computer model of robot arm

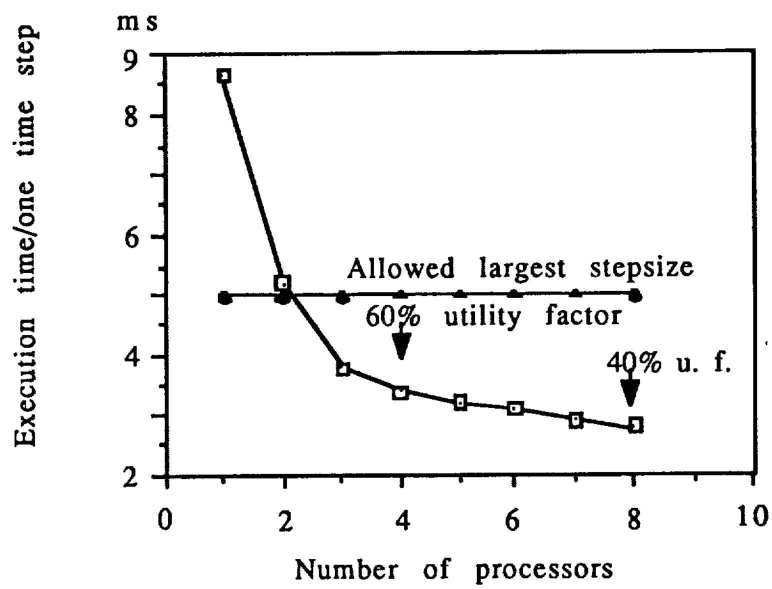

Figure 4. Execution time vs. number of processors for backhoe simulation

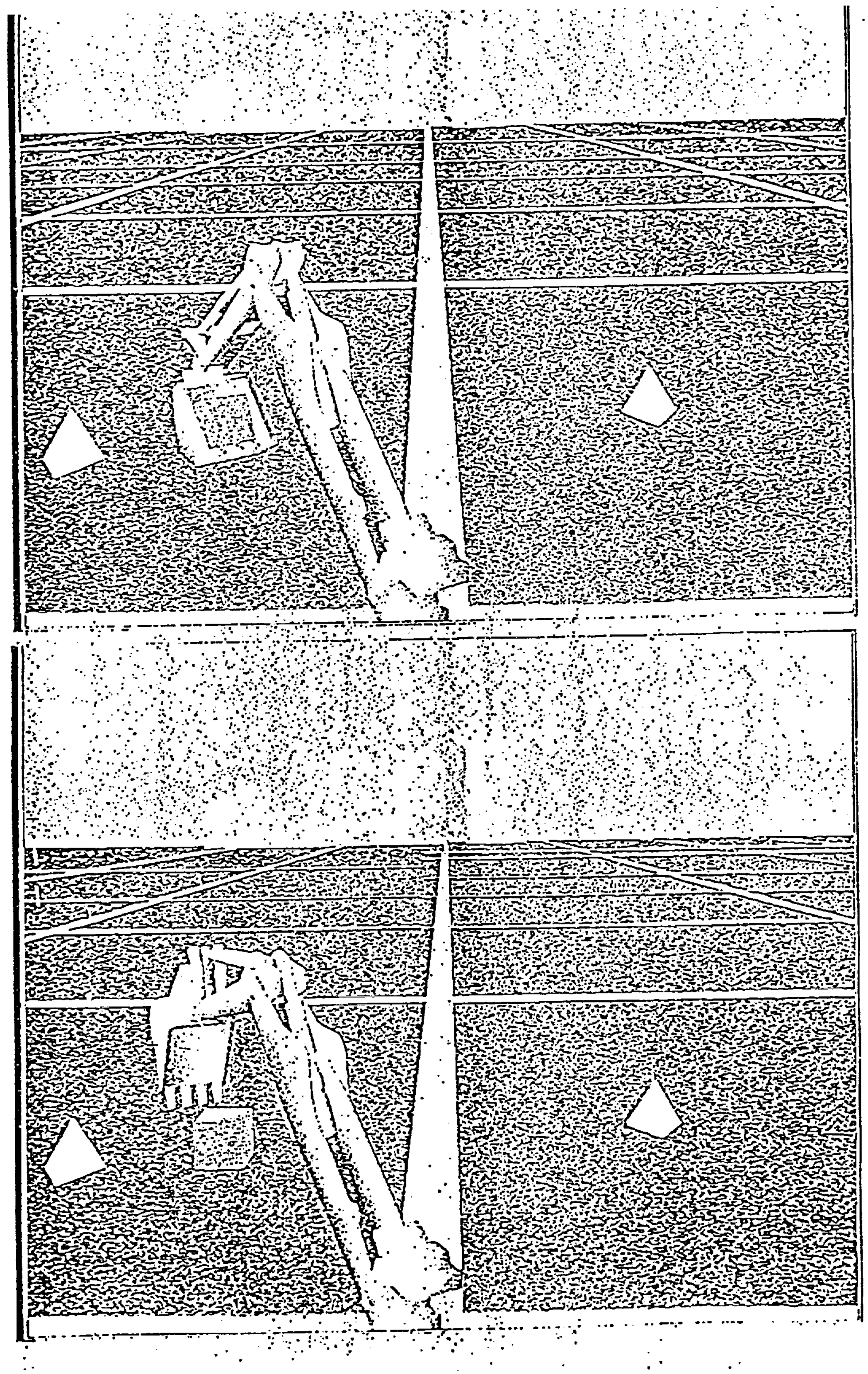

Figure 5. Interactive graphics display of backhoe

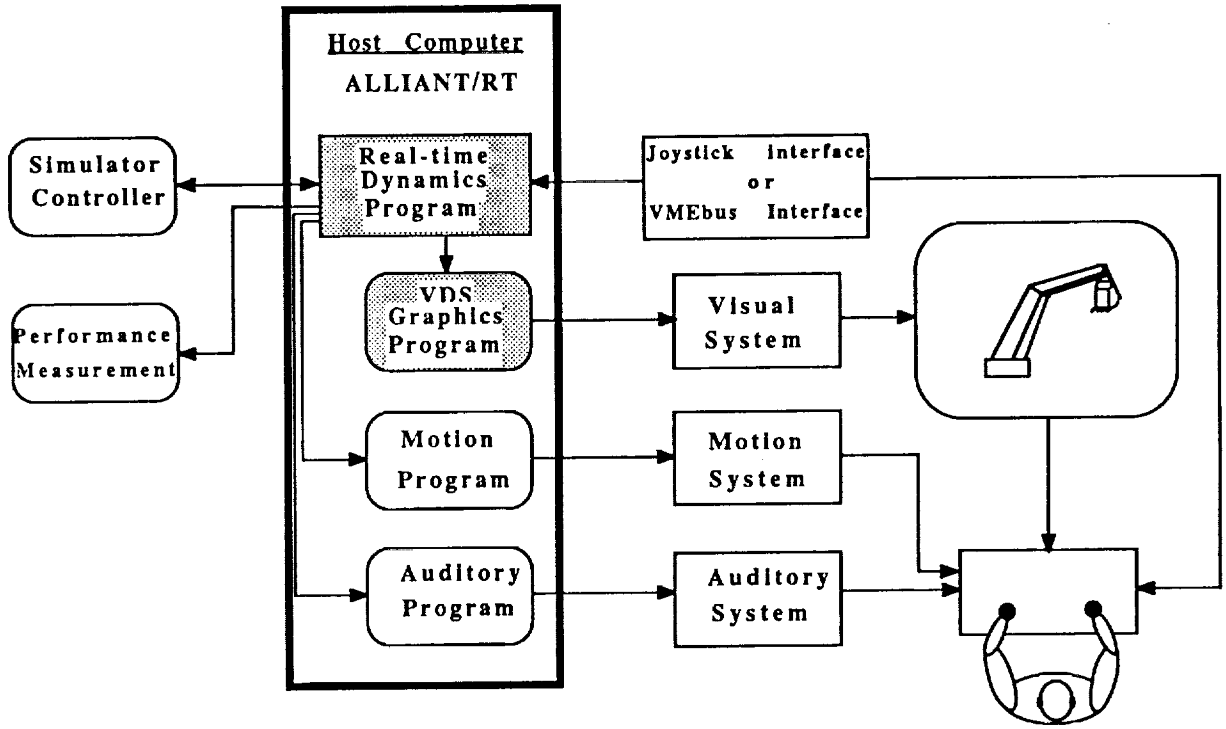

Figure 6. Workstation-based simulator